LLNL-TR-416495

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Calculation of damage, He and H production using SPECTER

J. Marian

September 2, 2009

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Calculation of damage, He and H production using SPECTER

Jaime Marian, LIFE Materials Group, MSTD/CMELS

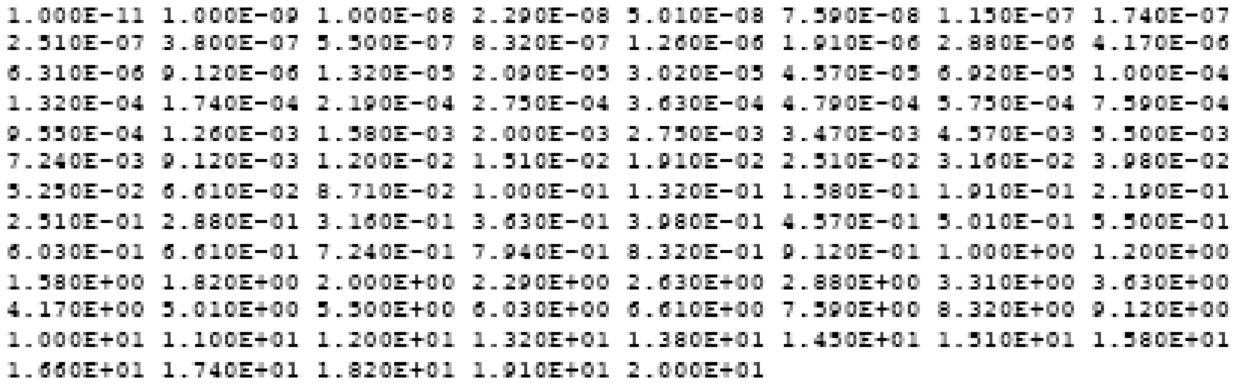
This document explains the procedure to compute gas production in fuel and structural materials using SPECTER from neutronics calculations provided by J. Latkowski using MCNP. The gas production cross sections (four channels are considered: $(n, {}^{1,2,3}_{1}H)$ and $(n, {}^{4}_{2}He)$) for each isotope are taken from the cross-section library ENDF/B-V (we refer the reader to the SPECTER manual for more details[1]). The calculation starts by examining the header of the Excel spreadsheet containing the flux data, in this case the vv2 scenario (60% fuel packing fraction) in the fuel region:

1. Filename	vv2000.o	vv2001.o	vv2002.o	vv2003.o	vv2004.o	vv2005.o	vv2006.o	…	
2. Time (y)	0.00	0.25	0.49	0.74	0.99	1.23	1.48	…	
3. Pfusion (MW)	500.00	500.00	500.00	500.00	500.00	500.00	500.00	…	
4. Source (n/s)	1.78E+20	1.78E+20	1.78E+20	1.78E+20	1.78E+20	1.78E+20	1.78E+20	…	
5. Fast flux (>100keV)	9.85E+13	1.21E+14	1.35E+14	1.48E+14	1.55E+14	1.64E+14	1.72E+14	…	n/cm^2-s
6. Interval fast fluence	7.66E+20	9.37E+20	1.05E+21	1.15E+21	1.20E+21	1.27E+21	1.34E+21	…	n/cm^2
7. Cumulative fast fluence	7.66E+20	1.70E+21	2.75E+21	3.90E+21	5.11E+21	6.38E+21	7.71E+21	…	n/cm^2
8. Total flux	3.82E+14	4.42E+14	4.82E+14	5.18E+14	5.36E+14	5.60E+14	5.80E+14	…	n/cm^2-s
9. Interval total fluence	2.97E+21	3.44E+21	3.75E+21	4.02E+21	4.17E+21	4.35E+21	4.51E+21	…	n/cm^2
10. Cumulative fluence	2.97E+21	6.41E+21	1.02E+22	1.42E+22	1.83E+22	2.27E+22	2.72E+22	…	n/cm^2

[1] SPECTER: Neutron Damage Calculations for Materials Irradiations, by L. R. Greenwood and R. K. Smither, ANK/FPP/TM-197

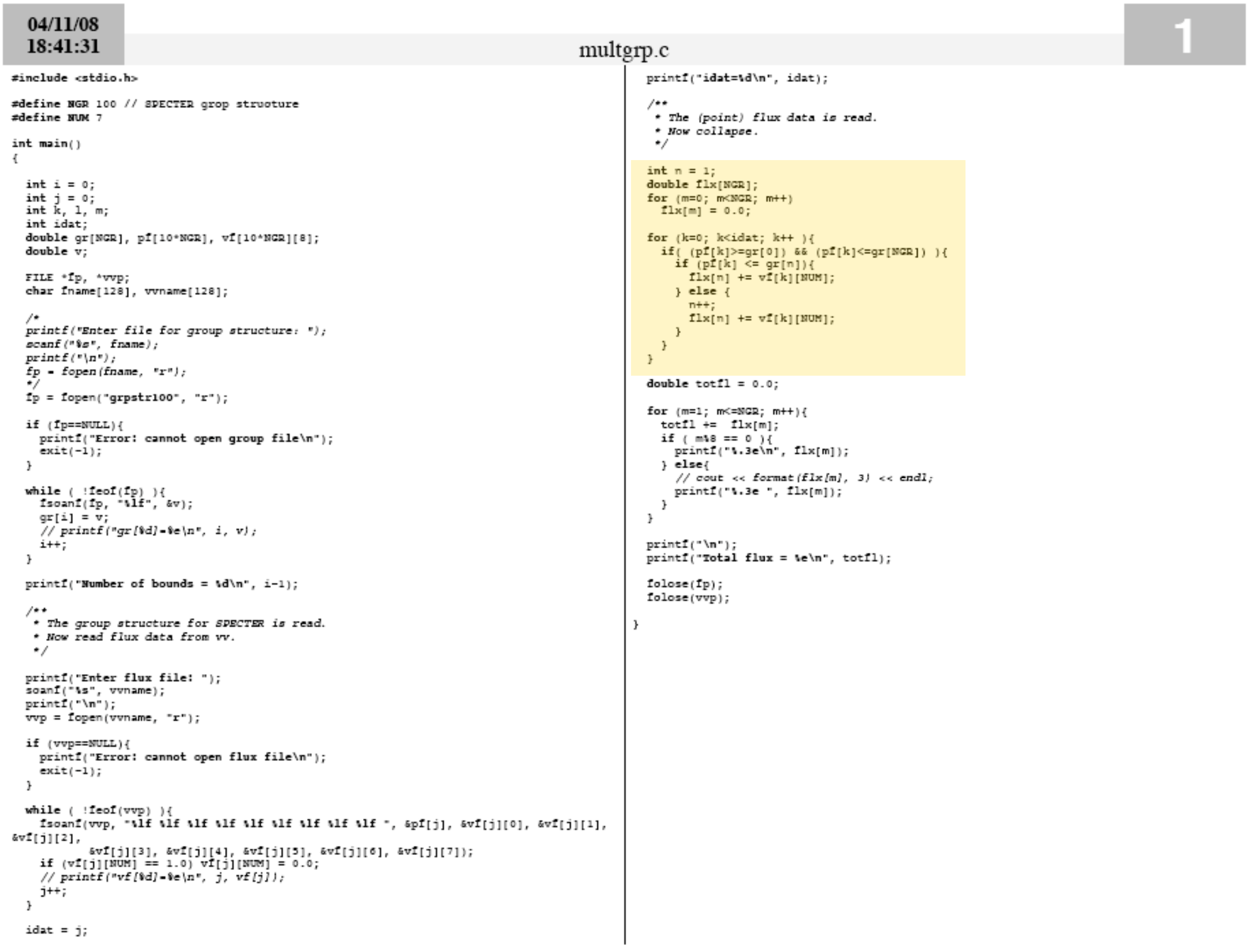
Therein, one can see flux data from, approximately, 0 (BOL) to 50 (EOL) years (cut after 7 columns for clarity). In order to compute damage and gas production with SPECTER, one needs to provide the flux group data expressed in a maximum of 100 energy groups (a SPECTER requirement).

The flux data structure provided by J. Latkowski using MCNP consists of 616 groups spanning the $10^{-9} < E < 20$ MeV energy range. The first thing that needs to be done is to collapse these data into a format consisting of 100 groups suitable for SPECTER. To this end, first we define the 101 energy bins as:



```
1.000E-11 1.000E-09 1.000E-08 2.290E-08 5.010E-08 7.590E-08 1.150E-07 1.740E-07
2.510E-07 3.800E-07 5.500E-07 8.320E-07 1.260E-06 1.910E-06 2.880E-06 4.170E-06
6.310E-06 9.120E-06 1.320E-05 2.090E-05 3.020E-05 4.570E-05 6.920E-05 1.000E-04
1.320E-04 1.740E-04 2.190E-04 2.750E-04 3.630E-04 4.790E-04 5.750E-04 7.590E-04
9.550E-04 1.260E-03 1.580E-03 2.000E-03 2.750E-03 3.470E-03 4.570E-03 5.500E-03
7.240E-03 9.120E-03 1.200E-02 1.510E-02 1.910E-02 2.510E-02 3.160E-02 3.980E-02
5.250E-02 6.610E-02 8.710E-02 1.000E-01 1.320E-01 1.580E-01 1.910E-01 2.190E-01
2.510E-01 2.880E-01 3.160E-01 3.630E-01 3.980E-01 4.570E-01 5.010E-01 5.500E-01
6.030E-01 6.610E-01 7.240E-01 7.940E-01 8.320E-01 9.120E-01 1.000E+00 1.200E+00
1.580E+00 1.820E+00 2.000E+00 2.290E+00 2.630E+00 2.880E+00 3.310E+00 3.630E+00
4.170E+00 5.010E+00 5.500E+00 6.030E+00 6.610E+00 7.590E+00 8.320E+00 9.120E+00
1.000E+01 1.100E+01 1.200E+01 1.320E+01 1.380E+01 1.450E+01 1.510E+01 1.580E+01
1.660E+01 1.740E+01 1.820E+01 1.910E+01 2.000E+01
```

As we can see, our (arbitrarily chosen) group structure covers the same energy range as the flux data from MCNP. A simple code (multgrp.c) has been written using the C programming language to do the collapse. This code only requires the desired output group structure as input (just shown above), and the flux data from MCNP. Note that the input flux group structure (from MCNP) can be any, as the code is not limited to the specific one of 616 groups used by Latkowski *et al.*. The code is shown here for whom might be interested:



04/11/08
18:41:31

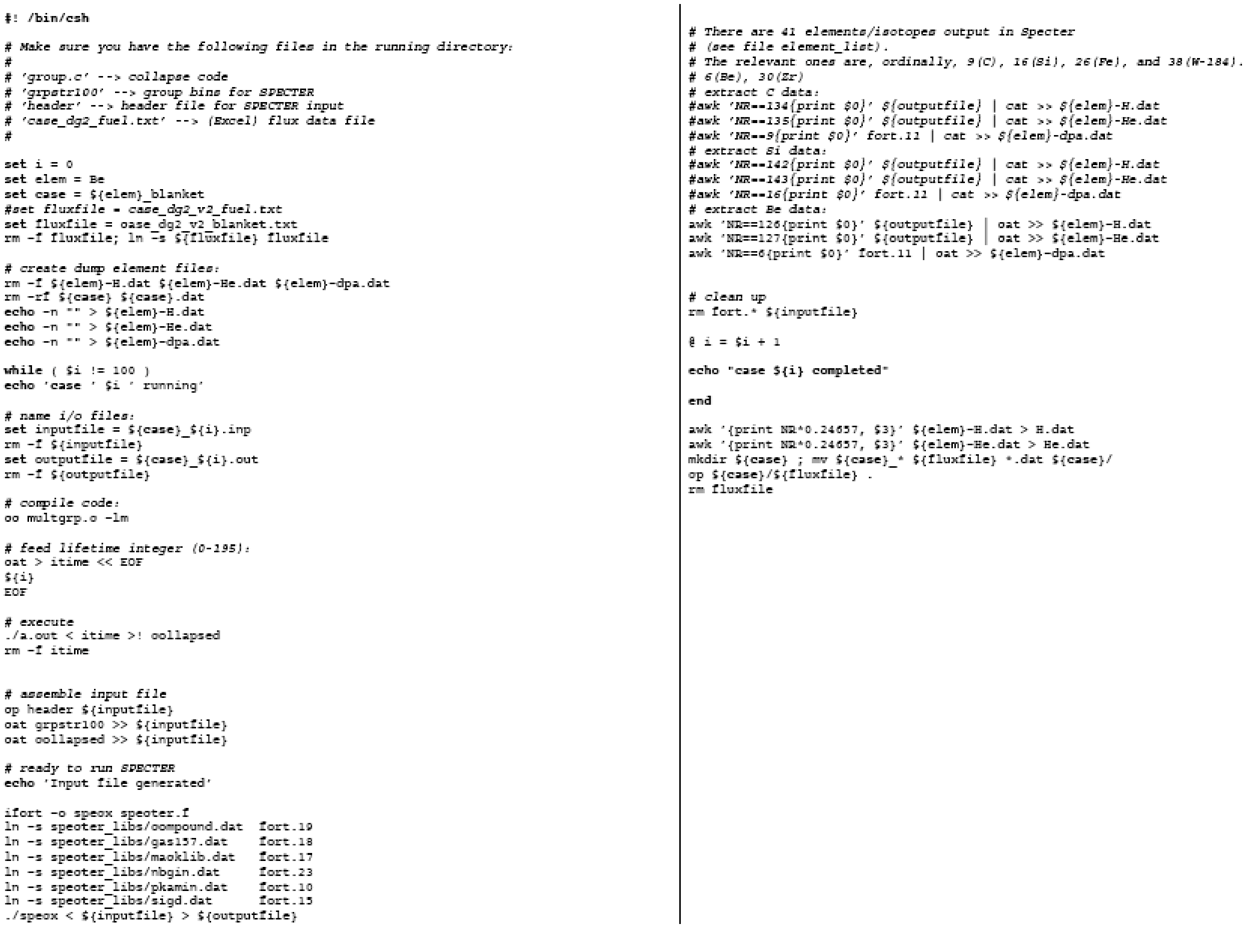
multgrp.c

1

```c
#include <stdio.h>

#define NGR 100 // SPECTER grop struoture
#define NUM 7

int main()
{

  int i = 0;
  int j = 0;
  int k, l, m;
  int idat;
  double gr[NGR], pf[10*NGR], vf[10*NGR][8];
  double v;

  FILE *fp, *vvp;
  char fname[128], vvname[128];

  /*
  printf("Enter file for group structure: ");
  scanf("%s", fname);
  printf("\n");
  fp = fopen(fname, "r");
  */
  fp = fopen("grpstr100", "r");

  if (fp==NULL){
    printf("Error: cannot open group file\n");
    exit(-1);
  }

  while ( !feof(fp) ){
    fsoanf(fp, "%lf", &v);
    gr[i] = v;
    // printf("gr[%d]=%e\n", i, v);
    i++;
  }

  printf("Number of bounds = %d\n", i-1);

  /**
   * The group structure for SPECTER is read.
   * Now read flux data from vv.
   */

  printf("Enter flux file: ");
  soanf("%s", vvname);
  printf("\n");
  vvp = fopen(vvname, "r");

  if (vvp==NULL){
    printf("Error: cannot open flux file\n");
    exit(-1);
  }

  while ( !feof(vvp) ){
    fsoanf(vvp, "%lf %lf %lf %lf %lf %lf %lf %lf %lf ", &pf[j], &vf[j][0], &vf[j][1],
&vf[j][2],
              &vf[j][3], &vf[j][4], &vf[j][5], &vf[j][6], &vf[j][7]);
    if (vf[j][NUM] == 1.0) vf[j][NUM] = 0.0;
    // printf("vf[%d]=%e\n", j, vf[j]);
    j++;
  }

  idat = j;
```

```c
  printf("idat=%d\n", idat);

  /**
   * The (point) flux data is read.
   * Now collapse.
   */

  int n = 1;
  double flx[NGR];
  for (m=0; m<NGR; m++)
    flx[m] = 0.0;

  for (k=0; k<idat; k++ ){
    if( (pf[k]>=gr[0]) && (pf[k]<=gr[NGR]) ){
      if (pf[k] <= gr[n]){
        flx[n] += vf[k][NUM];
      } else {
        n++;
        flx[n] += vf[k][NUM];
      }
    }
  }

  double totfl = 0.0;

  for (m=1; m<=NGR; m++){
    totfl +=  flx[m];
    if ( m%8 == 0 ){
      printf("%.3e\n", flx[m]);
    } else{
      // cout << format(flx[m], 3) << endl;
      printf("%.3e ", flx[m]);
    }
  }

  printf("\n");
  printf("Total flux = %e\n", totfl);

  folose(fp);
  folose(vvp);

}
```

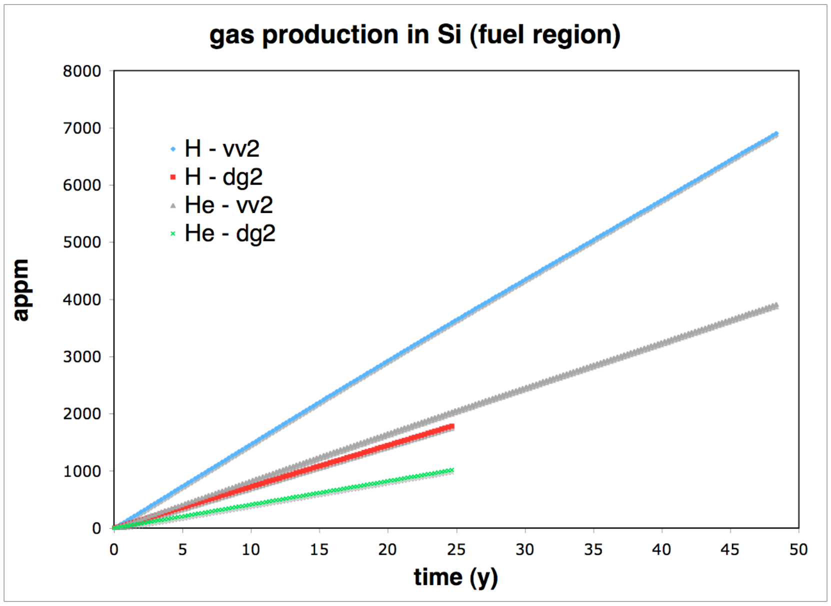
Essentially, multgrp.c contains the collapsing algorithm (highlighted) and support constructions for i/o and data manipulation. The output from this code is the collapsed flux in SPECTER format. This output has to be appended to a generic SPECTER input header to compose the final SPECTER input file. In this header is the other parameter that SPECTER needs before it is able to run, namely the irradiation time period over which results are to be calculated. As reported by Latkowski *et al.*, this interval is always 90 days. Once the input file is completely composed, SPECTER can then be run. The combination of input file generations, SPECTER runs and data extraction has been all integrated into a cshell script shown below:



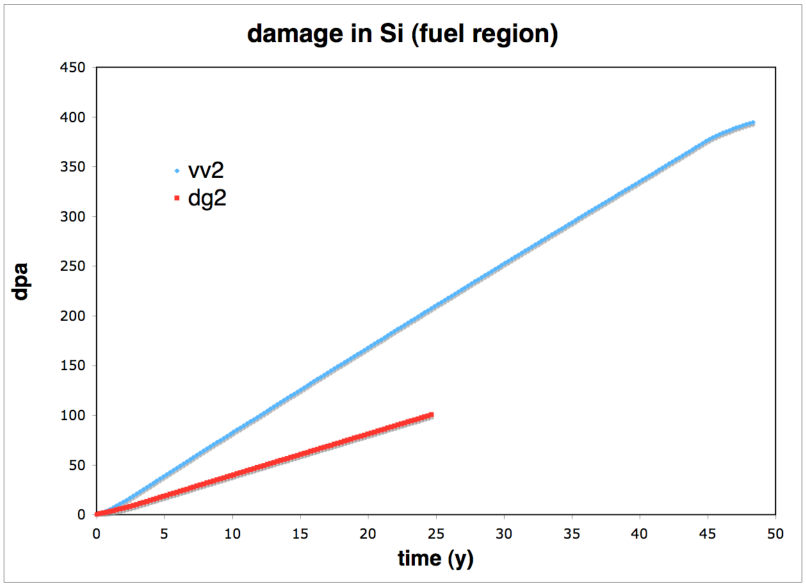
```
#! /bin/csh

# Make sure you have the following files in the running directory:
#
# 'group.c' --> collapse code
# 'grpstr100' --> group bins for SPECTER
# 'header' --> header file for SPECTER input
# 'case_dg2_fuel.txt' --> (Excel) flux data file
#

set i = 0
set elem = Be
set case = ${elem}_blanket
#set fluxfile = case_dg2_v2_fuel.txt
set fluxfile = case_dg2_v2_blanket.txt
rm -f fluxfile; ln -s ${fluxfile} fluxfile

# create dump element files:
rm -f ${elem}-H.dat ${elem}-He.dat ${elem}-dpa.dat
rm -rf ${case} ${case}.dat
echo -n "" > ${elem}-H.dat
echo -n "" > ${elem}-He.dat
echo -n "" > ${elem}-dpa.dat

while ( $i != 100 )
echo 'case ' $i ' running'

# name i/o files:
set inputfile = ${case}_${i}.inp
rm -f ${inputfile}
set outputfile = ${case}_${i}.out
rm -f ${outputfile}

# compile code:
cc multgrp.c -lm

# feed lifetime integer (0-195):
cat > itime << EOF
${i}
EOF

# execute
./a.out < itime >! collapsed
rm -f itime


# assemble input file
cp header ${inputfile}
cat grpstr100 >> ${inputfile}
cat collapsed >> ${inputfile}

# ready to run SPECTER
echo 'Input file generated'

ifort -o specx specter.f
ln -s specter_libs/compound.dat  fort.19
ln -s specter_libs/gas157.dat    fort.18
ln -s specter_libs/macklib.dat   fort.17
ln -s specter_libs/nbgin.dat     fort.23
ln -s specter_libs/pkamin.dat    fort.10
ln -s specter_libs/sigd.dat      fort.15
./specx < ${inputfile} > ${outputfile}
```

```
# There are 41 elements/isotopes output in Specter
# (see file element_list).
# The relevant ones are, ordinally, 9(C), 16(Si), 26(Fe), and 38(W-184).
# 6(Be), 30(Zr)
# extract C data:
#awk 'NR==134{print $0}' ${outputfile} | cat >> ${elem}-H.dat
#awk 'NR==135{print $0}' ${outputfile} | cat >> ${elem}-He.dat
#awk 'NR==9{print $0}' fort.11 | cat >> ${elem}-dpa.dat
# extract Si data:
#awk 'NR==142{print $0}' ${outputfile} | cat >> ${elem}-H.dat
#awk 'NR==143{print $0}' ${outputfile} | cat >> ${elem}-He.dat
#awk 'NR==16{print $0}' fort.11 | cat >> ${elem}-dpa.dat
# extract Be data:
awk 'NR==126{print $0}' ${outputfile} | cat >> ${elem}-H.dat
awk 'NR==127{print $0}' ${outputfile} | cat >> ${elem}-He.dat
awk 'NR==6{print $0}' fort.11 | cat >> ${elem}-dpa.dat


# clean up
rm fort.* ${inputfile}

@ i = $i + 1

echo "case ${i} completed"

end

awk '{print NR*0.24657, $3}' ${elem}-H.dat > H.dat
awk '{print NR*0.24657, $3}' ${elem}-He.dat > He.dat
mkdir ${case} ; mv ${case}_* ${fluxfile} *.dat ${case}/
cp ${case}/${fluxfile} .
rm fluxfile
```

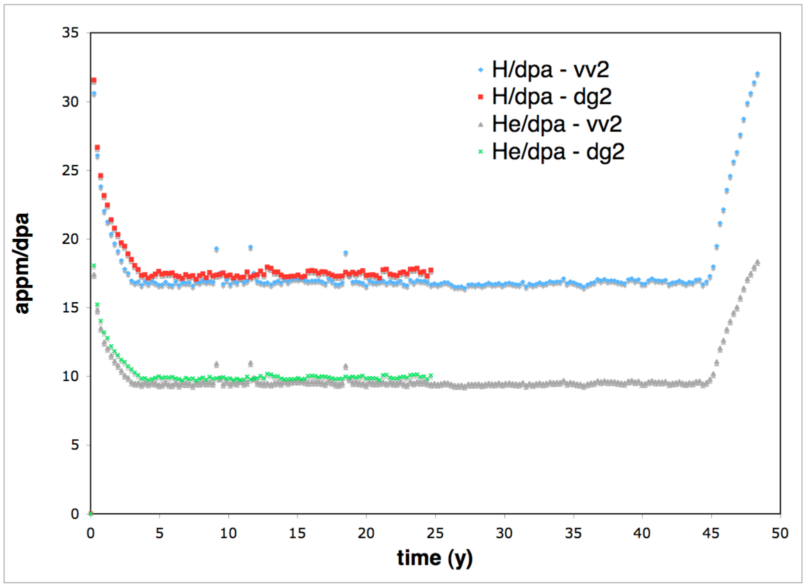
For Si in the fuel region we obtain the following time accumulation of H and He:



where results for the 30% packing fraction (flux case dg2) are also shown for comparison (note that time data for the dg2 case was only provided up to ~25 yrs.). We assume in these calculations that there is pure gas accumulation, i.e. that all the gas produced during a given time interval accumulates on top of that produced 'to-date' (no interval relaxation). The corresponding damage curve is:



Often, it is useful to know the gas-to-dpa ratios for each material so that particle irradiation –generally measured in dpa– can immediately be correlated with gas production. For Si in the fuel area, these ratios are:



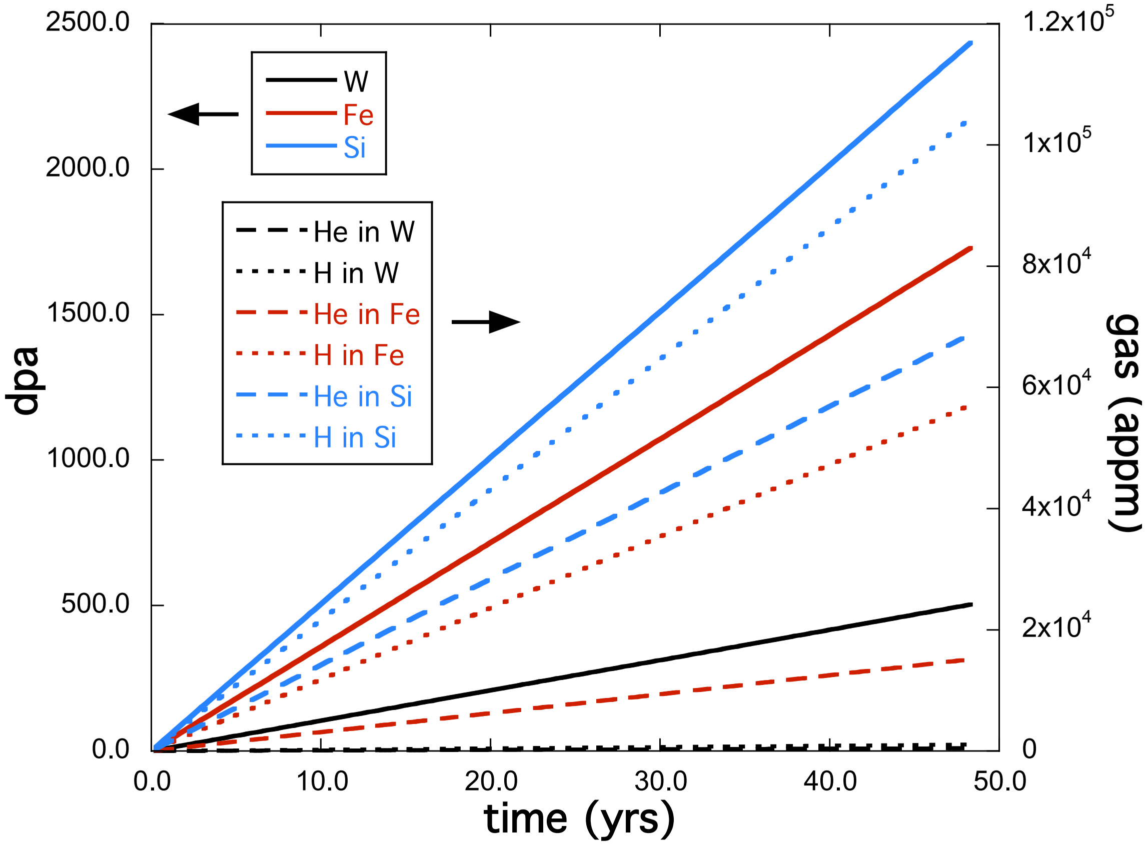
One can see that, although damage and gas accumulation results differ substantially, the gas-to-damage ratios are very close when going from 30 to 60% packing fraction. In calculating damage rates and accumulation, the critical parameter is the so-called threshold displacement energy (TDE), defined as the minimum energy necessary to knock an atom out of its position. SPECTER currently uses a value of 25 eV for covalent materials and of 40 eV for all transition metals (e.g., Si and Fe). These values simply are representative numbers that do not take into account the crystallographic orientation of the atomic collisions. However, it is now known that these dependencies may introduce variabilities of more than 100% in the TDEs. For example, Si TDEs have measured in the 30-80 eV range[2]. For its part, Fe TDEs have been found as low as 21 eV and as high as 53[3]. Fortunately, the values used by SPECTER are generally lower than the angle-averaged TDEs in most materials, leading to more conservative dpa estimates. The variability of TDEs in nuclear materials is nonetheless an important issue, and calculated dpa estimates must be considered keeping it in mind.

[2] L. Malerba and J. M. Perlado, *Phys. Rev*. B **65** (2002) 045202

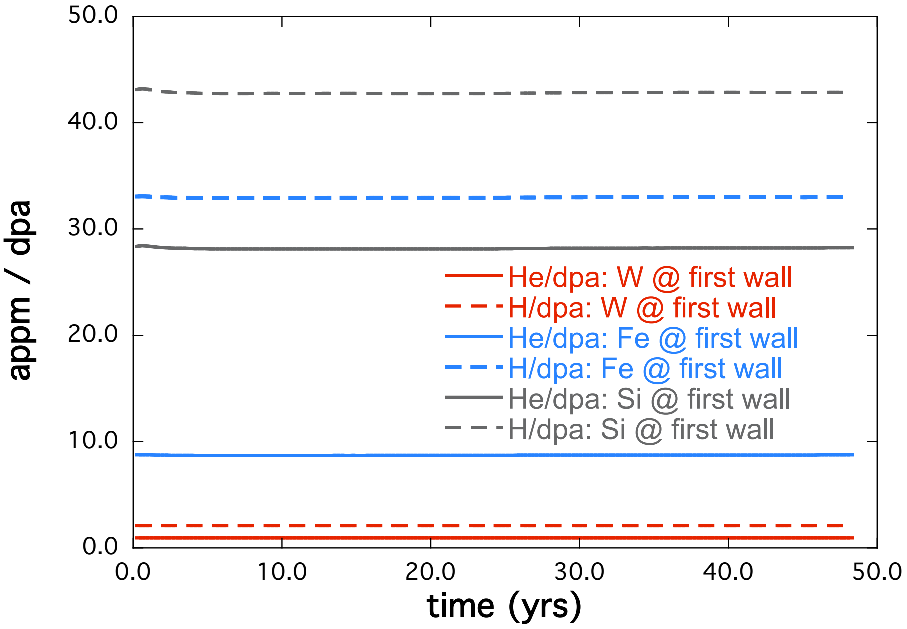
[3] K. Nordlund, J. Wallenius and L. Malerba, *Nucl. Instr. Meth. Phys. Res*. B **246** (2006) 322

First wall dpa and gas accumulation

We can straightforwardly repeat the same procedure for other materials of interest. For example, in the first wall, the damage and gas accumulation for W, Fe and Si are:

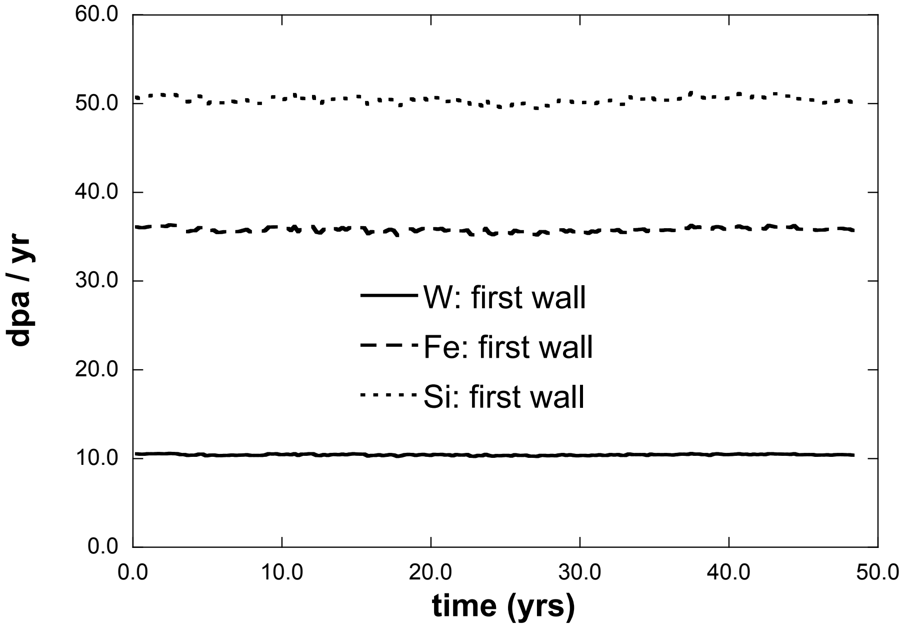


Here vv2 fluxes have been considered only (the more conservative choice). The gas-to-damage ratios for these materials are:

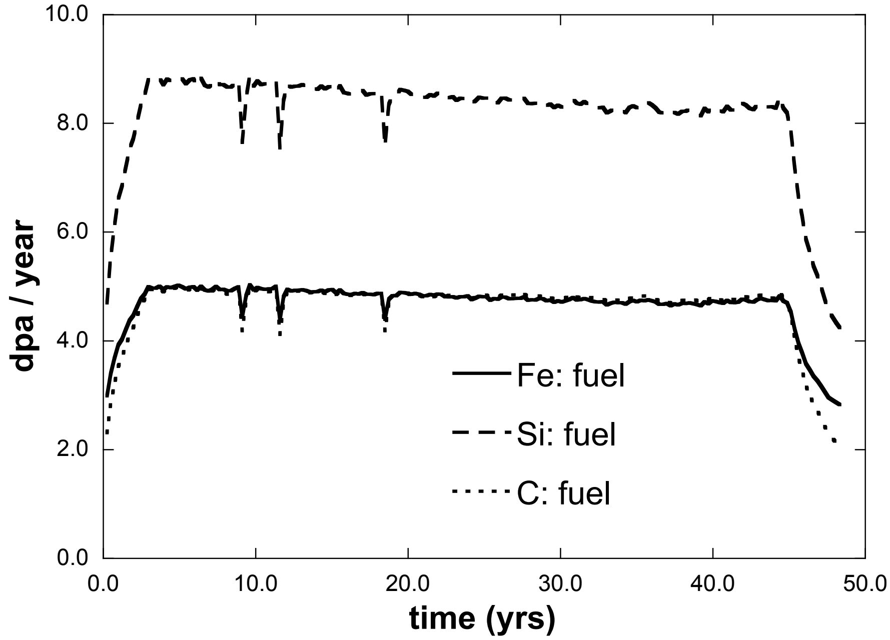


Damage rates

We can also look at the evolution of the dpa rate as a function of lifetime. For the first wall materials:

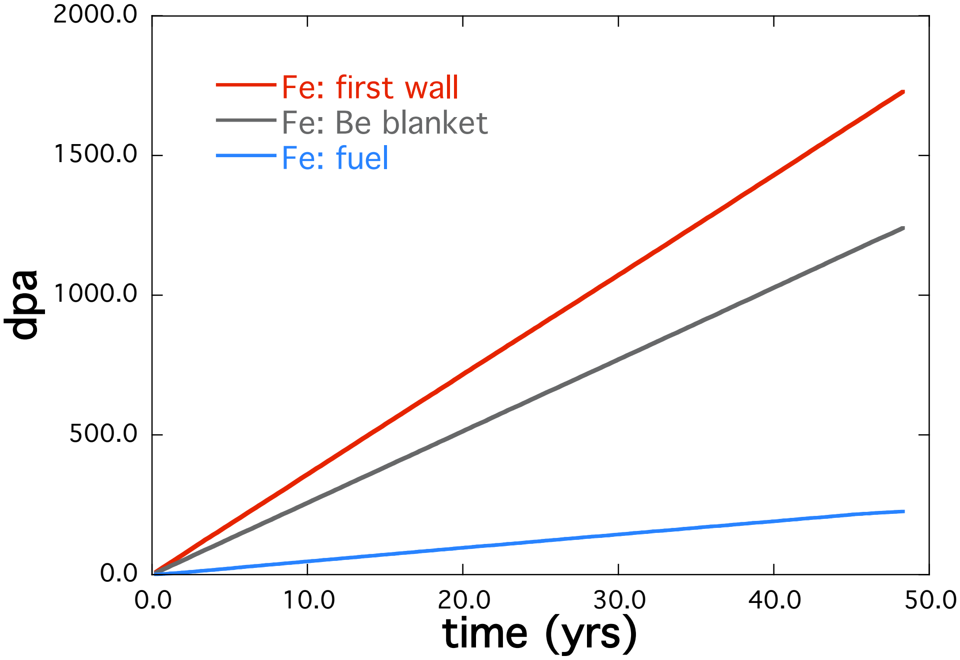


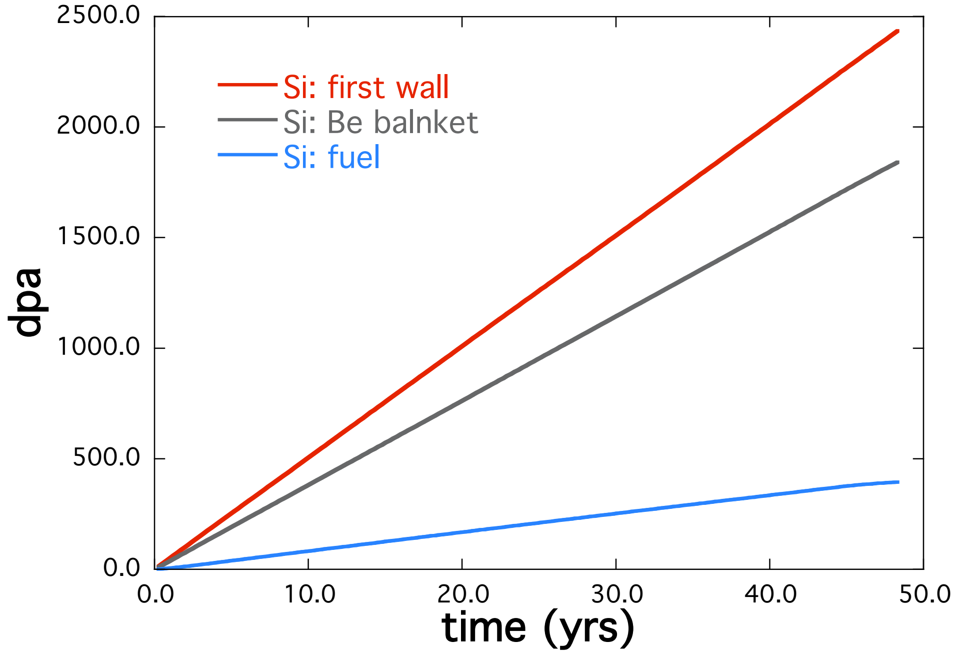
and in the fuel region:



Dpa accumulation as a function of lifetime and location for Fe and Si

Furthering the analysis, we can study the damage inflicted in Fe and Si as a function of the flux region (in other words, the radial coordinate):





In both cases (Fe and Si) we can see the drastic damage reduction when going from the first wall to the fuel region.

We can now express all these data into concise tables containing the critical information at 1 year of operation and for EOL for the materials of interest in each region. These are given in the following pages.

1-year, EOL data for first wall

Location:	First wall											
Time:	1 year						EOL (~50 yrs)					
element	dpa	He	H	dpa/yr	He/dpa	H/dpa	dpa	He	H	dpa/yr	He/dpa	H/dpa
W	10.4	9.8	21.8	10.5	0.95	2.1	503.7	478.9	1060.2	10.4	0.9	2.1
Fe	35.6	311.6	1176.3	36.1	8.8	33.1	1729.2	15135.9	57079.9	35.7	8.8	33.0

1-year, EOL data for fuel region

Location:	Fuel region											
Time:	1 year						EOL (~50 yrs)					
element	dpa	He	H	dpa/yr	He/dpa	H/dpa	dpa	He	H	dpa/yr	He/dpa	H/dpa
Si	5.7	87.6	138.9	6.6	15.5	24.5	394.9	4245.9	6806.4	4.3	10.7	17.2
C	2.93	274	0.0	3.48	78.4	0.0	223.9	12756	0.02	2.04	125	0.0

We emphasize that these results are for the case of 60% fuel packing fraction (vv2). Results for 30% (dg2) can be obtained easily if needed.